תרגיל 4
מגיש: רוי מור
מרצה: ד"ר אוסי מוקרין

שאלה מס׳ 1: שכבת התעבורה

למדנו כי פרוטוקול ה-UDP איננו אמין (unreliable) לעומת פרוטוקול ה-TCP שהוא אמין (reliable). מאידך, פרוטוקול ה-UDP איננו מצריך הקמת קשר טרם שליחת הנתונים (connectionless) ואילו TCP כן (connection oriented). עליכם לאפיין פרוטוקול חדש בשכבת התעבורה (Transport Layer) אשר ייקרא RUDP (Reliable UDP) ומטרתו לאפשר העברת נתונים אמינה שאיננה מצריכה הקמת קשר. הדבר דורש כמובן שנוי בפרוטוקול ה-UDP ולנוחיותכם מצורף מבנה פרוטוקול ה-UDP המקורי.

א. הגדירו מהי העברת נתונים אמינה

העברת נתונים אמינה מורכבת ממספר גורמים:

1. שמירה על תקינות תוכן ההודעה (אין איבוד או שינוי של המידע)
2. המידע מגיע באותו סדר שהוא נשלח
3. יכולת זיהוי שההודעה לא הגיע ושליחה חוזרת שלה

ב. הגדירו את מבנה הנתונים וצורת העבודה של שכבת התעבורה עבור RUDP. שימו לב, השינויים והתוספות אמורים להיות ישימים ובנוסף להתבסס על החומר שנלמד במהלך הקורס כך שאין צורך "להמציא" פתרונות מתוחכמים ו/או חדשים (פתרונות מהסוג האחרון לא יזכו במלוא הנקודות).

UDP Header

Source port	Destination port
UDP length	Checksum
Data (optional) …	

נגדיר מחדש את מבנה הנתונים וצורת העבודה של שכבת התעבורה עבור RUDP:

יהיה צורך להוסיף ל Header הקיים נתונים:

1. שדה אשר יהיה אחראי **למספור** של ההודעות - Seq
2. שדה אשר יהיה אחראי **לאישור** הודעות שהגיעו - Ack

צורת העבודה:

1. נסמן כל הודעה ע"י מספר מתאים (נתחיל ב 0), כאשר כל הודעות נוספת שלשלחת המספר משתנה בהתאם. כשאר ההודעה נשלחת יתחיל לפעול טיימר שיתאפס רק אחרי שתגיע תגובת ACK מתאימה מהצד השני(כלומר התגובה תכיל את ה Seq שאנו מצפים לו)
   *חשוב לציין כי לא תישלח הודעה נוספת עד לאישור קבלה להודעה האחרונה שנשלחה ושאין אפשרות להשתמש ב-ACK מצטבר: שדה ה-ACK/SEQ יכול לקבל רק 2 ערכים.
2. במקרה שלא התקבלה תגובה מתאימה עד מועד פקיעת הטיימר נשלח שוב את ההודעה המקורית עד לאישור קבלה. במקרה ונקבל שתי הודעות עם אותו מספר Seq נשלח את ההודעה בעלת ה ACK האחרון שקיבלנו אישור שהיא נשלחה בצורה תקינה.
   *ידוע לנו כי שכבות נמוכות יותר מ UDP ממשות מנגנונים לשמירה על תקינות המידע כגון, CRC ו – Checksum ולכן נניח שאין צורך לוודא את תקינות המידע שהגיע.

שאלה מס׳ 2: שכבת התעבורה

1) תהליך A שולח שני סגמנטים עוקבים של TCP לתהליך B. לסגמנט הראשון יש sequence number 90, ולשני sequence number 110.

a) כמה בתים של נתונים יש בסגמנט הראשון?

ניתן לראות כי המספר ברצף של הסגמנט הראשון הוא 90. שולחים כ – 20 בתים

מתקבל ACK שמספרו 110 (מחכים למידע מ110...) ולאחר מכאן המספר ברצף של הסגמנט הבא הוא 110 כפי שנתון.

b) נניח שהסגמנט הראשון הלך לאיבוד, אך השני הגיע ל-B. בהודעת ה-acknowledge ש-B ישלח ל-A, מה יהיה ה-acknowledgement number?

המספר אישור אותו נקבל יהיה 90.

(DUP ACK אשר נועד להגיע לשולח שחסר מידע בסדר הסגמנטים שהתקבל)

2) מה התפקיד של השדה ssthresh במנגנון ה- congestion avoidance? איך נקבע ערכו? מה הוא קצב גדילת החלון עד שמגיעים לערך זה, והאם הקצב משתנה כאשר מגיעים לערך זה? כיצד?

Slow Start Threshold – תפקידו לשמש נקודת סף ממנה מנגנון ה TCP עובר לעבוד במצב של congestion avoidance ממצב התחלתי של Slow Start.

כאשר ה congestion window קטן מ ssthresh שלב ה slow start פעיל וישנו קצב גידול **מעריכי** של החלון

ואילו במצב של congestion avoidance הופך קצב הגידול של ה congestion window

(שלב זה מתרחש כאשר congestion window > ssthresh) להיות **ליניארי.**

הערך של ssthresh נקבע להיות מחצית מה congestion window שהיה לפני האיבוד האחרון שהיה לשלוח,

כלומר נתקלנו ב3 DUP ACK רצופים או שהיה TIMEOUT בהמתנה ל ACK.

שאלה מס׳ 3: Network Layer – ידע כללי

א. נניח שיש שלושה נתבים בין מחשבי המקור והיעד. אם מתעלמים מפרגמנטציה, אזי חבילת IP שנשלחת ממחשב המקור למחשב היעד תעבור דרך כמה כרטיסי רשת? בכמה forwarding tables יחפשו כדי לדעת להיכן להעביר אותה?

1. החבילה יוצאת מכרטיס הרשת של מחשב המקור
2. החבילה מגיע לכרטיס הרשת של נתב I
   - הנתב בודק בטבלת הforwarding ומוציא את החבילה דרך הממשק המתאים (שדרכו ניתן להגיע ליעד)
3. החבילה יוצאת מכרטיס הרשת המתאים של נתב I
4. החבילה מגיע לכרטיס הרשת של נתב II
   - הנתב בודק בטבלת הforwarding ומוציא את החבילה דרך הממשק המתאים (שדרכו ניתן להגיע ליעד)
5. החבילה יוצאת מכרטיס הרשת המתאים של נתב II
6. החבילה מגיע לכרטיס הרשת של נתב III
   - הנתב בודק בטבלת הforwarding ומוציא את החבילה דרך הממשק המתאים (שדרכו ניתן להגיע ליעד)
7. החבילה יוצאת מכרטיס הרשת המתאים של נתב III
8. החבילה מגיע אל כרטיס הרשת של מחשב היעד

סה״כ עברנו 8 כרטיסי רשת (כולל שני המחשבים), ובחנו 3 טבלאות forwarding של הנתבים.

ב. נניח שאפליקציה מייצרת מקטעים של 40 בתים כל 20 msec, וכל מקטע נעטף (encapsulated) בכותרת (header) של סגמנט TCP ואז בכותרת של חבילת IP. איזה אחוז מהחבילה הוא תקורה, ואיזה הוא נתונים של האפליקציה?

ידוע לנו כי גודל ה encapsulation של ה headers של TCP הוא **20** וכי גודל ה encapsulation של ה headers של IP הוא **20 בתים** (ללא שימוש ב options).

סה״כ נקבל 40 בתים, נוסיף לזה את המידע עצמו אותו האפליקציה שולחת (40 בתים לפי הנתון) ונקבל חבילה בגודל 80 בתים. כלומר **50%** מתוך החבילה הסופית הם מידע ו**50%** נוספים הם overhead.

ג. הניחו שחבילת IP בגודל 3000 בתים נשלחת על לינק שלו MTU של 500 בתים, וה-ID שלה הוא 422. כמה פרגמנטים ייווצרו ומה יהיה בשדות הפרגמנטציה בכותרת של החבילה?

Fragment #	Length	ID	Offset	Frag
1	480	422	0	1
2	480	422	60	1
2	480	422	120	1
4	480	422	180	1
5	480	422	240	1
6	480	422	300	1
7	120	422	360	0
Total = 7				

*ניתן לראות כי לצורך העברת החבילה היא תחולק ל 7 פרגמנטים שונים.

שאלה מס׳4: Link Layer

א. מה ההבדל בין Slotted Aloha ל- Aloha? באיזה פרוטוקול יהיו פחות התנגשויות ולמה?

Aloha – כל תחנה יכולה לשדר בכל נקודת זמן
שלומר בשביל לשדר פריים שלם אנחנו צריכים שהתווח יהיה פנוי למשך שתי יחידות זמן.

Slotted Aloha – כל תחנה יכולה להתחיל לשדר רק בנקודת זמן קבוע
נדרש סנכרון בין התחנות וכתוצאה מזה מספיקה יחידת זמן אחת בשביל לשדר פריים שלם כי ברגע שהתחנה התחילה לשדר ולא הייתה התנגשות אז היא גם תסיים.
כיוון שאין שידור באמצע החלון אנו מבטלים למעשה את כל ההתנגשויות שנובעות מהתפרצות ובכך **קטנות מספר ההתנגשויות.**

ב. במה עדיף פרוטוקול CSMA על פרוטוקולי Aloha?

Carrier Sense Multi Access – CSMA כלומר יש יכולת ל״חוש״ את התווך – לדעת אם הוא בשימוש או לא ולכן יש לנו את האפשרות להימנע מהתנגשויות שנובעות מהתפרצות.

ג. מה ההבדל בין כתובות MAC לכתובות IP ?

כתובת IP

- כתובת לוגית בשכבת הרשת, ניתן לשנות אותה
- נועדה בשביל לממש היררכיה מסוימת
- מייצגת ישות ברשת והמיקום שלה (בתוך הרשת)
- גודל הכתובת 32 ביט

כתובת MAC

- כתובת שצרובה בכרטיס הרשת – ״כתובת פיזית״
- הכתובת מייצגת כרטיס רשת ממשי ולא ישות כמו במקרה של IP
- אין קשר היררכי בין שתי כתובות MAC
- בעזרת הכתובת אנו מבצעים מסירה של מידע בין שתי כרטיסי רשת
- גודל הכתובת 48 ביט

ד. בקובץ מעבדת Wireshark המצורף יש לענות על סעיפים: 1-4, 6-9, 12-16.

1. כתובת ה MAC של השולח 00:1f:d0:8d:06:e5
2. כתובת MAC של היעד f4:ec:38:b3:f6:d4
הכתובת שייכת לנתב שבעזרתו השולח יוצא מן הרשת מהקומית לעבר רשת היעד
3. הערך של שדה ה TYPE הוא 0x0800
הביטים מצביעים על כך שצריך להעביר את הטיפול לשכבת ה IP
4. ניתן לראות שמתחילת הפריים ועד להופעה האות G יש 54 בתים
6. כתובת ה MAC של השולח f4:ec:38:b3:f6:d4
הכתובת שייכת לנתב אשר מחבר את יוזם השיחה לשרת
7. כתובת MAC של היעד 00:1f:d0:8d:06:e5
הכתובת היא הכתובת של כרטיס הרשת במחשב של יוזם השיחה
8. הערך של שדה ה TYPE הוא 0x0800
הביטים מצביעים על כך שצריך להעביר את הטיפול לשכבת ה IP
9. ניתן לראות שמתחילת הפריים ועד להופעה האות O יש 67 בתים
12. כתובת המקור היא 00:1f:d0:8d:06:e5
כתובת היעד היא ff:ff:ff:ff:ff:ff
13. הערך של שדה ה TYPE הוא 0x0806
הביטים מצביעים על כך שצריך להעביר את הטיפול לפרוטוקול ה ARP
14. א. ניתן לראות כי לאחר 20 בתים מגיעים לשדה ה OPCODE
ב. הערך של השדה OPCODE הוא 0x0001 , מדובר בהודעה מסוג בקשה
ג. כן, בתוך המידע ניתן למצוא את השדה "Sender IP Address"
ד. השאלה "מהי כתובת ה MAC המתאימה לכתובת ה IP"
נמצאת ב4 הבתים אחרונים של בקשת ה ARP בשדה "Target IP Address"
15. א. ניתן לראות כי לאחר 20 בתים מגיעים לשדה ה OPCODE בתוך הנתונים של ה ARP
ב. הערך של השדה OPCODE הוא 0x0002 , מדובר בהודעה מסוג תשובה
ג. התשובה לשאלה נמצאת בתוך מידע ה ARP בשדה "Sender MAC Address"
16. כתובת המקור היא f4:ec:38:b3:f6:d4
כתובת היעד היא 00:1f:d0:8d:06:e5

שאלה מספר 5: Security and Cryptography

א. מהו יתרונה של שיטת DES על-פני RSA ?

הצפנה בעזרת מפתח ציבורי ב RSA היא פעולה שנגדירה אותה פעולה "כבדה" כיוון שהיא דורשת חישובים גדולים משמעותית (כמו כפל בשרשרת של מספרים גדולים) ואילו בהצפנה בשיטת DES אנו מצבעים פעולות שניתן להגדיר אותן "קלות" יחסית – כלומר במהירות גדולה (כמו חיבור או XOR)

לכן ניתן לראות שמבחינת זמן וכוח עיבוד ל DES יש יתרון אך ההצפנה ב RSA היא אכן מאובטחת יותר.

*כלומר אחרי ששני הצדדים יעבירו את המפתח בRSA נעדיף לעבור להשתמש ב AES\DES (בעזרת המפתח המשותף שעבר) שכן היא הרבה יותר יעילה כל זאת מבלי להתפשר על חוזק ההצפנה.

ב. אליס עושה שימוש בהצפנת מפתח ציבורי (public-key cryptography) לצרכים שונים. לשם כך היא רכשה מפתח, המורכב ממפתח פרטי (private key) ומפתח ציבורי (public key). כדי להגן על המפתח שלה מפני מזיקים, היא בחרה לשמור אותו במחשב שלה כאשר הוא מוצפן באמצעות סיסמה. את הסיסמה אליס זוכרת בעל-פה, אך היא חוששת שיום אחד תשכח אותה. לכן היא החליטה לשתף בסוד אדם נוסף, עליו היא סומכת באופן מוחלט. האדם הנבחר הוא עורך-הדין שלה, יובב, העושה אף הוא שימוש בהצפנת מפתח ציבורי.
מה אליס תשלח ליובב, ובאיזה אופן, כדי לשתף אותו באופן דיסקרטי בסוד המפתח שרכשה?

אליס צריכה לשלוח ליובב את המפתח הפרטי שלה (אותו היא רוצה לשתף איתו).

אם היא רוצה להעביר לו את המפתח באופן מוצפן כך שלא יהיה ניתן לפענח אותו היא צריכה להצפין את המפתח שלה בעזרת המפתח הציבורי של יובב וכמובן לשלוח את התוצאה ליובב.

אחרי שיובב יקבל את המפתח המוצפן הוא יוכל בעזרת המפתח הפרטי שלו לפענח\לחלץ את המפתח הפרטי של אליס ולשמור אותו לשעת צרה.

אין צורך לשלוח את המפתחות הציבורים כי מעצם הגדרתם הם ידועים לכל דורש.

בשפה מדוברת כלשהי יש רק 14 אותיות. טרודי תפס הודעה שנכתבה בשפה זו, והוצפנה בקוד הצבה פשוט (monoalphabetic substitution). השפה ושיטת ההצפנה מוכרים לו, וכעת הוא מעוניין לפצח את ההודעה, כלומר למצוא את המפתח לפיו בוצעה ההצבה. בספירה של מופעי האותיות, הוא גילה שנעשה שימוש רק ב- 11 אותיות שונות בגוף ההודעה. בהנחה שהוא יכול לפצח את ההודעה רק באמצעות ניסוי סדרתי של כל המפתחות (brute force), כמה ניסיונות יהיה עליו לבצע **בממוצע**, עד למציאת המפתח?

סה"כ ישנן 14! אפשרויות, נצמצם את מספר האפשרויות אשר לא מתקיימות כי נתון לנו שנעשה שימוש רק ב 11 אותיות (כלומר 3!) כלומר בממוצע יהיה עליו לעשות $14!/3!/2$ נסיונות עד למציאת המפתח. wolframalpha